

语义 Web 服务发现技术^①

王坤宁, 张翠肖

(石家庄铁道大学 信息科学与技术学院, 石家庄 050043)

摘要: 传统的 Web 服务发现技术大多采用基于关键字的服务匹配机制, 缺乏对 Web 服务功能的语义性描述, 影响了 Web 服务的查全率、查准率和查找的智能化。研究基于语义的 Web 服务匹配方法, 在经典 Web 服务匹配算法的基础上进行改进, 研究单概念语义 Web 服务匹配算法, 并由此得出多概念语义 Web 服务匹配的最优顺序, 结合输入、输出参数, 最终得出语义 Web 服务匹配结果, 实验结果表明, 改进后的方法比 BGM 算法匹配度高。

关键词: 语义 Web; 多概念语义 Web 服务; 匹配算法; Web 服务发现

Semantic-Based Web Service Discovery

WANG Kun-Ning, ZHANG Cui-Xiao

(School of Information Science and Technology, Shijiazhuang Tiedao University, Shijiazhuang 050043, China)

Abstract: The traditional Web services discovery technology mostly used keyword-based service matching mechanism, due to lack semantic Web services description, it affect the recall rate, precision rate and intelligent search of Web services. In this paper, we research on semantic-based Web service matching Algorithm. By improving the classical matching algorithm for Web services, study the concept of Semantic Web services single matching algorithm, and get the optimal order of concept of Multi-concept semantic Web service matching, then combining with the input and output parameters. Finally, get the results of semantic Web service matching.

Key words: semantic Web; multi-concept of semantic Web services; matching algorithm; Web services discovery

Web 服务发现实际上就是实现用户请求的服务和发布服务之间的匹配, Web 服务发现机制^[1]由三部分组成: 服务提供者、注册中心和服务请求者。三者间靠发布、查找和绑定等关系进行联系, 其中发布是由注册中心提供的接口, 主要作用是服务发布者提供将服务注册到注册中心的渠道; 查找提供查询接口, 以便应答服务请求者的请求, 提供请求者需要的服务; 绑定可以便于对服务提供者提供服务的调用。

目前已有的 Web 服务发现技术大多采用 Web 服务描述语言标准 WSDL(Web services Description Language)来描述 Web 服务, 采用统一描述、发现和集成标准 UDDI(Universal Description, Discovery and Integration)来注册和发现 Web 服务, 采用简单对象访问协议

SOAP(Simple Object Access Protocol)进行通信协议, 匹配算法采用了基于关键字的匹配算法^[2], 由于缺乏对 Web 服务功能的语义性描述, 从而导致“大数据量, 小信息量”的现象, 也无法实现自动的智能搜索。

因此, 提供基于语义的 Web 服务的发现机制, 以提高 Web 服务的查找效率成为研究热点。通过对现在的 Web 服务进行扩展, 定义和描述 Web 中的信息, 使之具有语义特征, 可以得出语义 Web 服务。因此, 人们对下一代互联网的构想主要集中在语义网^[3]方面, 它是一种可以被机器识别的文档集合, 集合了各种资源以及它们之间的语义关系。

文中就语义 Web 服务发现相关理论和技术, 在经典语义 Web 服务发现算法基础上进行改进。

^① 收稿时间:2011-12-13;收到修改稿时间:2012-02-19

1 经典的语义Web服务匹配方法

目前存在很多种语义 Web 服务匹配算法, 把 Web 服务抽象为包含语义关系的本体树是研究语义服务的一种思路. 文献[4]将 Web 服务用树形结构表示出来, 服务模型根节点、组合流程和原子流程分别代表树的根、中间节点和叶子节点. 文献[5]将服务模型实时构造成依赖图, 并提出 SAM 算法, 通过分析依赖图判读服务的可取性. 此方法划分的匹配粒度较为详细, 而且可以提示用户为部分匹配做出改进.

弹性匹配算法^[6]作为一种经典算法具有很好的指导意义, 使用离散值来评价某两个概念的匹配程度, 即将匹配结果按照其匹配程度划分为完全匹配、插拔匹配、包含匹配和匹配失败四类. 通过确定概念间的匹配程度, 提高了查准率和查全率, 是首次提出服务能力的匹配算法. 此文献具有很好的参考意义, 此后的很多学者在此基础上加以改进.

上述分类方法划分方式简单, 但是匹配等级比较少, 这样会造成无法对同等级内概念匹配程度做出更精确的划分. 因此, 本文在此基础上做了改进, 定量的计算服务请求和服务发布之间的匹配度, 这样得出的匹配度是一个介于 0 和 1 之间的实数.

2 单概念的语义匹配算法

在建立好本体之后, 概念间通过继承关系和属性关系联系起来, 这种关联性越大, 说明概念间语义匹配度越高, 反之说明语义匹配度越低. 通过分析对比, 本文采用的是基于概念距离的相似度计算方法, 通过生成节点路径表的方法计算单概念语义距离和相似度, 并在经典相似度计算方法 BGM 算法的基础上进行改进, 通过单概念匹配获取最优顺序, 然后计算多概念语义匹配相似度, 以提高匹配率.

2.1 权重分配

权重分配指对本体概念间关系的权重分配. 若用 dep 表示在本体树中所处的深度, 用 sub 表示概念间为继承关系, 用 prop 表示概念间为属性关系, 假设 d1 为 d2 的子概念, d1 和 d2 的相似度与 d2 所处的深度成反比, 则继承关系权重如公式(1)所示:

$$Q[\text{sub}(d1,d2)]=1/(\text{dep}(d2)) \quad (1)$$

若 d1 为 d2 的属性, d1 与 d2 的相似度与概念属性所处的深度为减函数关系, 则属性关系权重如公式(2)

所示:

$$Q[\text{prop}(d1,d2)]=n*(1/f(\text{dep}(d2))) \quad (2)$$

其中, n 为大于 1 的实数, 这里 f(dep(d))为与概念 d 所处的本体树深度有关系的函数, 这里取 $f(\text{dep}(d)) = k^{\text{dep}(d)}$.

2.2 节点路径表的生成

如果本体概念个数较多, 则会大大增加语义距离计算的时间, 降低效率, 因此本文采用的是生成节点路径表的方法, 首先把本体中的每一个节点和其根节点的路径和路径中节点间的权重记录下来, 计算时直接从已经建立好的节点路径表中提取.

生成节点路径表的具体方法为: 从根节点广度遍历, 每个子节点的路径为其父节点的路径和本节点构成, 以此类推. 假设 d1 和 d2 为本体中的两个概念, 且 d1 为 d2 的子节点, dc 为根节点, 用 L 表示概念间的路径, 则 d1 到根节点 dc 的路径 $L(d1,dc) = L(d2,dc) + L(d1,d2)$.

2.3 计算得出语义距离

假设 d1 和 d2 为本体中的两个概念, dc 为根节点, 用 L 表示概念间的路径, 用 distance 表示两概念间的距离, 用 Q 表示概念间的权重, 语义距离的计算分为以下几种情况:

①若 d1 和 d2 为同一个概念, 则两概念间距离为 0, 即 $\text{distance}(d1,d2)=0$;

②若 d1 和 d2 有直接联系, 则比较 d1 到 d2 的距离和 d2 到 d1 的距离, 取两者中较小的值, 即 $\text{distance}(d1,d2)=\min\{Q[L(d1,d2)], Q[L(d2,d1)]\}$;

③若 d1 和 d2 之间有间接联系, 则首先找出 d1 和 d2 之间的路径, 再把 d1 和 d2 之间的距离分解为两两直接联系的距离之和, 即 $\text{distance}(d1,d2)=\text{distance}(d1,dn \cdots .dm,d2)=\sum\{Q[L(d1,dn)] \cdots . Q[L(dm,d2)]\}$, 若存在多条路径, 则以最短路径为准;

④若 d1 和 d2 之间没有间接关系, 则两概念之间的距离为 d1 和 d2 分别到根节点的最短距离之和. 即 $\text{distance}(d1,d2)=\min\{\text{distance}(d1,dc)\}+\min\{\text{distance}(d2,dc)\}$.

2.4 计算得出相似度

相似度的计算是以语义距离为基础的, 即相似度一定是语义距离的函数, 相似度需要满足以下要求:

①当两个概念间的语义距离为 0 时, 相似度为 1;

②两概念间的相似度随着两者间的语义距离的增加而减小, 反之亦然;

③两概念间的相似度是介于 0 和 1 之间的实数. 但是满足以上条件的函数很多, 比如: $sim1=1/(1+distance)$, $sim2=1/(distance^2+1)$, $sim3=1/(distance+1)^2$, $sim4=1/e^{distance}$. 经过分析得知, 四个函数下降速度依次增加, 指数下降的速度最快, 本文选择 $sim1$ 作为相似度函数. 因为函数的下降速度会因不同的本体而不同, 因此需要在此函数中引入参数 m , 即 $sim=1/(1+m*distance)$, m 为介于 0 与 1 之间的实数.

现有本体树, 结构图如 1 所示. 在此图中, 以实线连接的两个概念为继承关系, 以虚线相连的为属性关系.

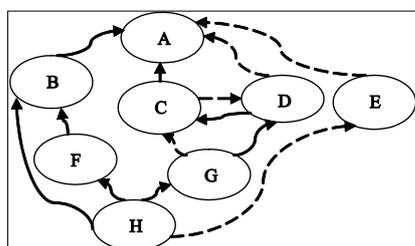


图 1 单概念匹配本体

各概念间的权重值如表 1. 其中 n 取 2, 因图中的本体平均深度为 4, 因此 k 取 2.

表 1 概念间的权重值

边	权重	边	权重
B->A	1	G->C	1
C->A	1	G->D	0.5
D->A	2	C->D	1
E->A	2	D->C	0.5
H->B	0.5	H->G	0.25
F->B	0.5	H->E	1
H->F	0.25	—	—

各个节点到根节点的路径及其权重如表 2 所示.

表 2 节点到根节点的路径及权重

边	路径 L	权重 Q	边	路径 L	权重 Q
B->A	B,A	1	E->A	E,A	2
C->A	C,A	1	H->A	H,B,A	0.5,1
D->A	D,A	2	H->A	H,F,B,A	0.25,0.5,1
F->A	F,B,A	0.5,1	H->A	H,G,C,A	0.25,1,1
G->A	G,C,A	1,1	H->A	H,G,D,A	0.25,0.5,2
G->A	G,D,A	0.5,2	H->A	H,G,C,D,A	0.25,1,1,2
G->A	G,C,D,A	1,1,2	H->A	H,G,D,C,A	0.25,0.5,0.5,1
G->A	G,D,C,A	0.5,0.5,1	H->A	H,E,A	1,2

以 C 节点为例, 计算得出 C 节点与其它节点的语义距离, 结果如表 3 所示. 表中的所属情况一栏表示 C 节点与其他节点的语义距离计算过程属于前面介绍的语义距离计算的第几种情况.

表 3 C 节点与其它节点的语义距离

节点与节点	所属情况	语义距离	节点与节点	所属情况	语义距离
C->A	(2)	1	C->E	(4)	3
C->B	(4)	2	C->F	(4)	2.5
C->C	(1)	0	C->G	(2)	1
C->D	(2)	0.5	C->H	(3)	1.25

以 C 节点为例, 计算得出 C 节点与其它节点的相似度, 结果如表 4 所示. 计算过程中取公式 $sim=1/(1+m*distance)$, 其中 m 取 0.5.

表 4 C 节点与其它节点的相似度

节点与节点	相似度	节点与节点	相似度
C->A	0.667	C->E	0.4
C->B	0.5	C->F	0.444
C->C	1	C->G	0.667
C->D	0.8	C->H	0.615

3 语义Web服务的输入输出匹配

在多概念语义匹配算法中, 首先需要利用单概念相似度算法计算得出每个输入/输出的概念相似度, 然后通过对此结果进行的求和运算, 得出最终的语义相似度.

BGM(Balance Genealogy Measure)算法是比较经典的多概念相似度算法, 它在文献[7]的基础上加以改进, 不再要求待匹配的概念必须为叶子节点. 当存在重复匹配的情况时, 会降低该元素与另一元素的相似度. BGM 算法的匹配最优顺序是由推理获取的, 而且匹配顺序不同, 匹配值的大小也不相同.

假设有两个集合 $H1, H2$, 分别对应的树为 $T1, T2$, 若要计算这两个集合的相似度, BGM 算法的步骤如下:

1)使用最优顺序访问 $T1$ 中的每个叶子 $l1$;

2)在 $T2$ 中找到与 $l1$ 最佳匹配的节点 $l2$, 若 $T2$ 中有多个叶子与 $l1$ 均为最佳匹配, 则选择被匹配次数最少的节点作为最佳匹配节点, 并将此节点的匹配次数加 1, 即 $count(l2)$ 加 1;

3)用同样的方法把 T1 中的所有叶子节点匹配完。

本文的算法在此基础上加以改进,在最优顺序获取时,并不是通过推理获取,而是通过前面介绍的单概念算法计算得出概念间的相似度,并按大小顺序依次排列,以供选取。改进后的算法如下:

1)输入匹配。在输入匹配过程中,请求的输入应该满足候选服务的输入,此时的匹配是具有方向性的。若用 $inputr$ 和 $inputa$ 表示请求服务的输入集合和候选服务的输入集合,则两者的相似度如公式(3)所示

$$S(inputa, inputr) = \frac{\sum_{I1 \in inputa} S(I1)}{Ninputa} \quad (3)$$

式中, $S(I1)$ —— $inputa$ 中的概念 $I1$ 与 $inputr$ 匹配的相似度; $Ninputa$ ——候选服务的输入概念集合 $inputa$ 的大小。 $S(I1)$ 如公式(4)所示:

$$S(I1) = S(I1, I2) * \beta^{\text{count}(I2)-1} \quad (4)$$

式中, $I2$ —— $inputr$ 中与 $I1$ 相似度最高且匹配次数最少的元素; $S(I1, I2)$ —— $I1$ 和 $I2$ 的相似度; $\text{count}(I2)$ —— $inputr$ 中的元素 $I2$ 被 $inputa$ 中元素最佳匹配过的次数。其中 $S(I1, I2)$ 可以通过单概念语义匹配算法得出。

2)输出匹配。在输出匹配过程中,服务的输出应该满足请求的输出,此时的匹配是具有方向性的。服务输出与服务输入相似度计算过程相似,用 $outputr$ 和 $outputa$ 分别表示请求服务的输出集合和候选服务的输出集合,则两者的相似度如公式(5)所示:

$$S(outputr, outputa) = \frac{\sum_{I1 \in outputr} S(I1)}{Noutputr} \quad (5)$$

式中, $S(I1)$ —— $outputa$ 中的概念 $I1$ 与 $outputr$ 匹配的相似度; $Noutputr$ ——服务请求输出概念集合 $outputr$ 的大小。 $S(I1)$ 如公式(6)所示:

$$S(I1) = S(I1, I2) * \beta^{\text{count}(I2)-1} \quad (6)$$

式中, $I2$ —— $outputr$ 中与 $I1$ 相似度最高且匹配次数最少的元素; $S(I1, I2)$ —— $I1$ 和 $I2$ 的相似度; $\text{count}(I2)$ —— $outputr$ 中的元素 $I2$ 被 $outputa$ 中元素最佳匹配过的次数。其中 $S(I1, I2)$ 可以通过单概念语义匹配算法得出。

在输入匹配和输出匹配相似度计算完成之后,就可以得出综合匹配度 Sim , 如公式(7)所示。

$$Sim = N1 * siminput + N2 * simoutput \quad (7)$$

式中, $siminput$ ——输入参数相似度; $simoutput$ ——输出参数相似度。其中 $N1$ 和 $N2$ 为根据实际情况或经验值设定的权值,符合条件 $N1+N2=1$ 。

4 结语

如何从大量的 Web 服务中快速准确的找到合适的 Web 服务研究具有重要意义,而传统的 Web 服务匹配方法已不能满足用户对系统效率的要求,文中对经典的语义 Web 服务算法——BGM 算法进行改进,通过单概念语义 Web 服务匹配算法得出多概念语义 Web 服务匹配的最优顺序,并结合输入、输出参数进行多概念语义 Web 服务匹配。

参考文献

- 1 Paolucci M, Kawamura T, Payne TR, Sycara K. Semantic Matching of Web Services Capabilities. Lan Horrocks, James Hendler. The Semantic Web-ISWC 2002. Berlin: Springer Verlag, 2002, 279-291.
- 2 Tsai TM, Yu HK, Liao PY, et al. Semantic Modeling among Web Services Interfaces for Services Integration-SOTA (Smart Office Task Automation) Platform. Vladimir Marik, Werner Retschizegger, Olga Stepankova. Database and Expert Systems Applications. Washington: IEEE Computer Society 2003, 2003.579-583.
- 3 张敏.基于语义网的学术论文检索的研究与实现.沈阳:沈阳工业大学,2011.
- 4 Bansal S. Matchmaking of Web Services Based on the DAML-S Service Model. Adelaide: University of South Camlina, 2002.
- 5 Brogi A, Corfini S, Popeseu R. Composition-oriented Service Discovery. University of Pisa: Department of Computer Science, 2005.
- 6 Paolucci M, Kawamura T, Payne TR, et al. Semantic Matching of Web Services Capabilities. Lan Horrocks, James Hendler. The Semantic Web-ISWC 2002. Berlin: Springer, 2002. 333-347.
- 7 Sharma A, Adarkar H, Sengupata. Managing QoS through Prioritization in Web Services. Proc. of the 4th International Conference on Web Information Systems Engineering Workshops. Washington: IEEE Computer Society Press, 2003. 140-148.